



AI's Next Leap In Redefining Enterprise Software Development: From Past to Future Possibilities

Author: Syed Younus

Abstract

Software Development as a field is exponentially growing. It is deemed essential to innovate and lead. Though technologies from time to time have greatly impacted the development methodologies, it is AI now that has ushered software development into a new era of possibilities. It truly has helped with optimizing the SDLC (Software Development Lifecycles), but there are myriad challenges that its integration into software engineering entails. AI tools are fast transforming the ways the codes are written, generated, improved, optimized, reviewed, debugged, and managed. From saving time that often goes into carrying out repetitive tasks to analyzing or testing code, AI tools have made almost all things simple. NLP (Natural Language Processing) is one of the best core features of AI-human interaction. The users provide queries in simple English and get the results they need to speed up their development process. This paper highlights the current practices, strategies, methodologies, and procedures enterprises employ to speed up their development processes. It systematically reviews the existing literature and discusses the difficulties underlying AI integration into enterprise software development practices, the complex nature of the AI models that the companies use, data storage, and data security. Further, it analyzes the AI-human collaboration from different perspectives, and how it fosters a culture of continuous learning,

enabling the developers to learn and grow their development skills too. It also sheds light on the use of the current practices and what the future possibilities are.

Keywords: Software Development, Software Engineering, AI in Software Development, Artificial Intelligence in Software Engineering.

Introduction

The term software has myriad definitions (Khan, 2021). It is defined as a system that is capable of performing different tasks (Fitzpatrick, 1996). It could be a product that is born out of new and innovative ideas with a focus on resolving a problem (Mohagheghi, et al., 2009). It is defined as a system that can help minimize human effort with the power to deliver tasks at a rapid speed (Schott and Ovtcharova, 2024).

Selecting the Right Software Programming Language

Companies, whether small or big, know the significance of building software. It could be for internal teams or the customers. A product is developed following a set of procedures and rules. Different types of programming help with turning ideas into new software products such as JavaScript, Java, Rust, Typescript, Cotlin, C#, Carbon, V Lang, etc. It is noteworthy to mention that each of these



software development languages has its limitations (Yang, et al., 2024). An idea that can be developed in Kotlin might not be possible to take the form of software if another language is employed, say, C or C+. What is possible to develop in Java is not possible to develop in other languages? Selecting a proper language for the development is itself a big challenge for the consultants and the developers (Yazymov, et al., 2024). But, with the introduction of AI into software development consulting, it has become much simpler. The consultants provide the inputs and describe how the software should be and the AI tool can instantly generate the possible solutions (Becker and Gottschlich, 2021). It can provide clear steps on how one can implement the software ideas, what language could be helpful to develop the ideas into a solution, and how challenges could easily be overcome during the process of development (Barenkamp, et al., 2020).

Code Writing and Code Generation

Alenezi and Akour (2025) stress that writing code is a thing of the past. Many of the companies prefer to generate the code rather than writing the code. Ever since the beginning of software development, writing code has been a normal process (Donohoe, 2012). An employee working as a programmer or a coder needs to have a good understanding of the code. He/she had to spend a lot of time understanding the requirements and then producing the code that helped with resolving the problem at hand. Experience in development was a key to successful development. An employer would prefer to hire only those employees who have great coding skills. But, currently, the scenario is different. Now, employers prefer to employ people, who have enough skills in integrating

AI with the SDLC. Even a developer with substantial knowledge of coding can work on developing the program with assistance from AI tools (Askarov, 2022). The use of AI plays a key role in optimizing almost all core facets of the SDLC. For instance, code generation is now the term used to generate the code. Writing the code from scratch is a thing of the past. The AI tool is capable enough to generate the required based on the queries (Padmanaban and Sharma, 2019).

SDLC and AI

Software development is a time-consuming process. It takes a lot of time. It involves investment. There could be risks. There could be failures. A correct development strategy plays a key role in developing the software as per the defined goals (Sunil, 2025). Software development projects fail due to inappropriate planning. When human attention dwindles or flickers, the project deliverables are disrupted (Hourani, et al., 2019).

The best thing about generating the code is of course NLP. The users can interact with the system in simple plain English. The system will produce the code the way the users want (Shin and Nam, 2021). For instance, It can show results for all types of scripts from simple to advanced. Users can get a script for a basic Python-based calculator, a script to design and develop a Java-based to-do list, or any simple and more productive scripts for CRUD application (CRUD (Create, Read, Update, Delete), web scraping, and other types of programs (Dehaerne, 2022).

The system will also suggest the next code structure that will align with the previous one. It will autocomplete the code or allow its users to change or modify the code depending on the



software requirements. Not just this, it will also review the given code, and rephrase or optimize it for better clarity and function (Kamble, et al., 2024)

Debugging With AI - Satisfactory or Not?

Bug Detection was one big challenge. It requires teamwork plus time. Testing the software from different perspectives became a little less hectic with the AI integration into bug detection systems. The system will analyze the code and find out if any potential bugs might affect the performance of the software. It will play a key role in generating the test cases to enhance software reliability (Pandy, et al., 2024). The users streamline software testing not by writing the code or modifying the code on their own but only with the proper instructions provided by the AI tool. The tool will perform debugging based on the queries and directions provided or it will also suggest directions to move forward which will further streamline the debugging process.

But, of course, relying completely on the AI output for debugging may not yield better results. As it is a very important aspect of software development, it needs to be more strategic with a human team in the loop. The AI system will produce the reports but analyzing the reports carefully and taking precautionary measures to perform debugging actions from different angles where human teams will take on the charge is very important (Khan, et al., 2024).

Project Management with AI

Project management is one big process and the way the projects are managed determines the success or the failure of the project. If the project is strategically planned, executed,

supervised, and managed then it is deemed to be successful but if there are errors or loopholes in the management process then it will lead to big failures. Software projects are budget-intensive and therefore a small error could lead to big losses. Managing the project with the power of AI is one core aspect of the SDLC. Organizations have realized how important it can be to use AI in managing all types of programming-related projects and how it could also save time and money. Everything from defining project goals to developing business cases, creating a more suitable work breakdown, developing the project schedule, tracking the work done by the employees, estimating resources, budgets, risks, and other core aspects of management is now AI-powered (Odejide and Edunjobi, 2024).

Apart from this, most of the AI systems today come with advanced capabilities of predictive analytics. These innovative tools can predict what can lower the performance of the software, and what bugs might appear based on the historical data (Shoushtari, et al., 2024). It takes into account the projects done, and the coding structure of the previous similar types of programs and provides necessary guidance and suggestions (Kiani, 2024). Apart from identifying the sources of the error, it moves to the next level where it provides the tools and the resources to enhance the project management process. Of course, all these tasks are done using NLP (Natural Language Processing) which is one of the best core features of AI-human interaction.

Decision-Making With AI

One of the most transformative approaches to improving software development is AI-driven Decision-making. Leveraging the power of



insights obtained from the data, companies can streamline their enterprise-grade operations (Coussement, et al., 2024). Integration of analytics into almost all processes boosts the project's success. Small and big companies use analytics as a strategic move to develop enterprise-grade solutions for their clients. Data from the past on almost all different types of projects with different entities such as timelines, teams that worked on, resource utilization, errors, testing processes, and other core processes, help companies successfully identify patterns and become powerful enough to avoid similar kinds of mistakes in the future (Anand and Miglani, 2024). Machine learning techniques and statistical data analysis make it easy for enterprises to extract actionable insights and arrive at a better decision. Though decision-making becomes more simplified with insights, the companies need to implement precautionary measures and incorporate human understanding and human-based analysis too. Lessons learned from past projects, successful or failed projects, make companies more adaptable and they can strengthen their position effectively well in innovating, and developing new kinds of solutions that are flawless (Biesialska, et al., 2021).

No doubt, software projects are well beyond investment risk. They are more prone to susceptible changes, any change at any time, and any disruption in the development process. Dealing with these unforeseen difficulties intimidates even the big firms. But, with timely reports on analytics, companies overcome this problem. For instance, analytics can help define the project scope, and how much time a project will take to complete based on historical evidence. This report can be useful for the team leaders and they can scale up and down their time as required to

complete the project on time. This way companies save themselves from potential risks or any bad impact (Rahamathunnisa, et al., 2023).

Team Collaboration in Real-time

Companies see a great increase in team productivity when AI is integrated into team collaboration and team management. AI helps with improving communication, enhancing the overall success rate of the project. Teams can collaborate, post their queries, and get answers to their questions. Human-AI Collaboration works when both AI systems and human-only teams are synced well following a proper strategy (Sunil, 2025). Organizations need to identify and analyze their limits and give more credence to human intelligence rather than just the output produced by AI systems. For instance, AI systems could err in producing the right code and the human-only teams need to be vigilant enough to bring this to the board where members of the board can check and resolve the issue. Similarly, errors by human-only teams analyzed and detected by the AI systems should be cross-checked and verified. Limitations in Human-AI collaboration still exist but the benefits it offers are huge and seem to increase in the future too (Ebert and Louridas, 2023).

Smart Investment and AI

Software development practices backed up with the power of artificial intelligence deliver great results. companies that use AI in software development practices see a significant surge in the quality of the product and also save time. One of the core challenges in implementing the AI system into the software development process is the investment. which could be huge (Palomo, et al., 2021). A skilled team with great



capabilities to make it big for the companies could be hard to find. But yes, the talent hunting should be strategic and such great teams should be built. Companies operating within a tight budget could not be in a position to implement AI or may prefer to go for a third-party AI system that could align with their software development project and assist their teams.

Furthermore, there could be challenges in terms of dealing with the resistance to change, or use of the AI in development. Training programs, workshops, and other similar kinds of events that could help build and elevate AI knowledge and its use in software programming could be more helpful (Deshmukh, 2023).

Many companies resist the change because of the huge investment it entails. But, to overcome these problems, companies can take on a step-by-step approach. They can implement this new AI-driven strategy in phases. This way they will not have to set up infrastructure, hire the resources, or manage teams. They can start with one phase of development and when they see good results they can move to the next level (Matt, 2024). For instance, they can integrate AI systems into their software development practices to generate the code. which will facilitate the team in building the software projects super fast. The AI tool will continue to generate the code at the request and make the task easier for the developer Even a developer with little knowledge can improve his understanding of the coding working with the AI system.

Conclusion

Artificial Intelligence has certainly boosted software development practices, making it

easy for development teams, project leaders, and stakeholders to get insights into every facet of development and also build wonderful software solutions in less time. The ease of coding, code reviewing, code optimization, and testing have improved greatly with the AI. AI's transformative capabilities certainly have helped teams coordinate and collaborate more effectively, identify and resolve problems, and manage the products by seamlessly leveraging insights and reports. However, it is well-productive when it is paired with human intelligence since software development is risky and entirely dependent upon the AI without a human-only team it will lead to great losses. Though AI is a boon for the modern software development industry it is with great caution that teams or organizations should employ or use it and should prioritize AI-based code structuring, code requirement, testing, and other core processes with due human intelligence The future of whether AI will continue to dominate the world or replace the human teams is still bleak but it is possible that it will at least reduce the team size significantly and organizations will tend to employ the leaders who could work parallel with the AI teams.

References

- Khan, F., Kumar, R. L., Kadry, S., & Nam, Y. (2021). The future of software engineering: Visions of 2025 and beyond. *International Journal of Electrical and Computer Engineering*, 11(4), 3443-3450.
- Fitzpatrick, R. (1996). Software quality: definitions and strategic issues.
- Mohagheghi, P., Dehlen, V., & Neple, T. (2009). Definitions and approaches to model quality in model-based software



development—A review of literature. *Information and software technology*, 51(12), 1646-1669.

Schott, V., & Ovtcharova, J. (2024). Integrating Sustainability into Software Development: A Global Categorization. In *Finance and Law in the Metaverse World* (pp. 59-69). Springer, Cham.

Yang, H., Nong, Y., Wang, S., & Cai, H. (2024). Multi-Language Software Development: Issues, Challenges, and Solutions. *IEEE Transactions on Software Engineering*.

Yazymov, M., Yarashova, G., & Atayev, A. (2024). MODERN INNOVATIVE TECHNOLOGIES IN THE FIELD OF SOFTWARE. *Всемирный ученый*, 1(29), 286-293.

Becker, K., & Gottschlich, J. (2021, July). AI Programmer: autonomously creating software programs using genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1513-1521).

Barenkamp, M., Rebstadt, J., & Thomas, O. (2020). Applications of AI in classical software engineering. *AI Perspectives*, 2(1), 1.

Alenezi, M., & Akour, M. (2025). AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Applied Sciences*, 15(3), 1344.

Donohoe, P. (Ed.). (2012). Software product lines: Experience and research directions.

Askarov, E. (2022). ARTIFICIAL INTELLIGENCE AND SOFTWARE BASED

ARTIFICIAL INTELLIGENCE. *Oriental renaissance: Innovative, educational, natural and social sciences*, 2(12), 722-728.

Padmanaban, P. H., & Sharma, Y. K. (2019). Implication of Artificial Intelligence in Software Development Life Cycle: A state of the art review. *2019 IJRRRA all rights reserved*.

Sunil Medepalli. (2025). AI In Modern Software Development: Current Practices, Challenges, and Future Possibilities. In *AI In Modern Software Development: Current Practices, Challenges and Future Possibilities*. (Vol. 10, Number 01, pp. 1–7). Zenodo. <https://doi.org/10.5281/zenodo.14716031>

Hourani, H., Hammad, A., & Lafi, M. (2019, April). The impact of artificial intelligence on software testing. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)* (pp. 565-570). IEEE.

Shin, J., & Nam, J. (2021). A survey of automatic code generation from natural language. *Journal of Information Processing Systems*, 17(3), 537-555.

Dehaerne, E., Dey, B., Halder, S., De Gendt, S., & Meert, W. (2022). Code generation using machine learning: A systematic review. *Ieee Access*, 10, 82434-82455.

Kamble, N., Khode, P., Dhabekar, V., Gode, S., Kumbhare, S., Wagh, Y., & Mohod, D. S. (2024). Code Generation Using AI and NLP Based Technique. *Available at SSRN 4824877*.

Pandy, G., Pugazhenth, V. J., & Murugan, A. (2024). Advances in software testing in 2024: Experimental insights, frameworks, and future directions. *International Journal of Advanced*



Research in Computer and Communication Engineering, 13(11), 40-44.

Khan, M. F. I., Mahmud, F. U., Hoseen, A., & Masum, A. K. M. (2024). A new approach of software test automation using ai. *Journal of Basic Science and Engineering*, 21(1), 559-570.

Odejide, O. A., & Edunjobi, T. E. (2024). AI in project management: exploring theoretical models for decision-making and risk management. *Engineering Science & Technology Journal*, 5(3), 1072-1085.

Shoushtari, F., Daghighi, A., & Ghafourian, E. (2024). Application of Artificial Intelligence in Project Management. *International journal of industrial engineering and operational research*, 6(2), 49-63.

Kiani, A. (2024). Artificial intelligence in entrepreneurial project management: a review, framework and research agenda. *International Journal of Managing Projects in Business*.

Coussement, K., Abedin, M. Z., Kraus, M., Maldonado, S., & Topuz, K. (2024). Explainable AI for enhanced decision-making. *Decision Support Systems*, 114276.

Anand, S., & Miglani, S. (2024). *Real-time AI-driven predictive analytics for agile software development: Enhancing decision-making, resource optimization, and risk mitigation*.

Biesialska, K., Franch, X., & Muntés-Mulero, V. (2021). Big Data analytics in Agile software development: A systematic mapping study. *Information and Software Technology*, 132, 106448.

Rahamathunnisa, U., Subhashini, P., Aancy, H. M., Meenakshi, S., & Boopathi, S. (2023). Solutions for Software Requirement Risks Using Artificial Intelligence Techniques. In *Handbook of Research on Data Science and Cybersecurity Innovations in Industry 4.0 Technologies* (pp. 45-64). IGI Global.

Sunil Medepalli. (2025). Human-AI Collaboration (HAIC): The Rise of Hybrid Intelligence in Modern Software Development. In Human-AI Collaboration (HAIC): The Rise of Hybrid Intelligence in Modern Software Development (1.00, Vol. 10, Number 1, pp. 1–6). Zenodo. <https://doi.org/10.5281/zenodo.14743406>

Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. *IEEE Software*, 40(4), 30-38.

Palomo-Duarte, M., García-Domínguez, A., & Balderas, A. (2021). Assessment in software development for competitive environments: An AI strategy development case study. *Electronics*, 10(13), 1566.

Deshmukh, A., Patil, D. S., Mohan, J. S., Balamurugan, G., & Tyagi, A. K. (2023). Transforming Next Generation-Based Artificial Intelligence for Software Development: Current Status, Issues, Challenges, and Future Opportunities. In *Emerging Technologies and Digital Transformation in the Manufacturing Industry* (pp. 30-66). IGI Global.

Matt, D. (2024). *The implementation of AI technology in international companies: a leadership perspective* (Doctoral dissertation, FH Vorarlberg (Fachhochschule Vorarlberg)).

